package com.twitter.frigate.pushservice.util

import com.twitter.finagle.stats.BroadcastStatsReceiver

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.frigate.pushservice.model.PushTypes.Target

import com.twitter.frigate.pushservice.thriftscala.PushResponse

import com.twitter.frigate.pushservice.thriftscala.PushStatus

import com.twitter.frigate.thriftscala.CommonRecommendationType

object ResponseStatsTrackUtils {

def trackStatsForResponseToRequest(

crt: CommonRecommendationType,

target: Target,

response: PushResponse,

receivers: Seq[StatsReceiver]

)(

originalStats: StatsReceiver

): Unit = {

val newReceivers = Seq(

originalStats

.scope("is\_model\_training\_data")

.scope(target.isModelTrainingData.toString),

originalStats.scope("scribe\_target").scope(IbisScribeTargets.crtToScribeTarget(crt))

)

val broadcastStats = BroadcastStatsReceiver(receivers)

val broadcastStatsWithExpts = BroadcastStatsReceiver(newReceivers ++ receivers)

if (response.status == PushStatus.Sent) {

if (target.isModelTrainingData) {

broadcastStats.counter("num\_training\_data\_recs\_sent").incr()

}

}

broadcastStatsWithExpts.counter(response.status.toString).incr()

if (response.status == PushStatus.Filtered) {

broadcastStats

.scope(response.status.toString)

.counter(response.filteredBy.getOrElse("None"))

.incr()

}

}

}